AD 700 816

# ON THE CLOSURE OF AFL UNDER REVERSAL

Seymour Ginsburg, et al

System Development Corporation
Santa Monica, California

November 1969

TM-738/056/00





SCIENTIFIC REPORT
NO. 2

# On the Closure of AFL Under Reversal

Seymour Ginsburg and Michael Harrison

10 November 1969

Contract Monitor: Thomas G. Evans Data Sciences Laboratory

Monitored by: Lt. Col. Russell Ives, SRIR (AFOSR) Supported by:

SDC's Independent Research Program:

Contract F1962870C0023, Programming (Algorithmic) Languages,

Project No. 5632, Task No. 563205, and Grant No. AF/AFOSR/1203/67A

Work Unit No. 56320501

27

Prepared for
AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH, USAF
BEDFORD, MASSACHUSETTS 01730
and
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
OFFICE OF AEROSPACE RESEARCH, USAF
ARLINGTON, VIRGINIA

Qualified requestors may obtain additional copies from the Defense Documentation Center. All others should apply to the Clearinghouse for Federal Scientific and Technical Information.

AFCRL-69-0551

TM-738/056/00

SDC
TM SERIES

The work reported herein was supported in part by Contract F1962870C0023, CRL -- Algorithmic Languages, Project No. 5632, Task No. 563205, Work Unit No. 56320501; by Grant No. AF-AFOSR-1203-67A; and by NSF Grant No. GJ454.

SCIENTIFIC REPORT NO. 2

10 November 1969

# ON THE CLOSURE OF AFL UNDER REVERSAL

by
Seymour Ginsburg*
and
Michael Harrison**

Monitored by: Contract Monitor
Thomas G. Evans
Data Science Laboratory
Major Russell Ives, USAF
SRIR (AFOSR)

Prepared for:
AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH, USAF
BEDFORD, MASSACHUSETTS 01730
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
OFFICE OF AEROSPACE RESEARCH, USAF
ARLINGTON, VIRGINIA

SYSTEM
DEVELOPMENT
CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA
90406



*System Development Corporation and University of Southern California

**System Development Corporation and University of California, Berkeley

## ABSTRACT

A simple sufficiency condition is given for an abstract family of acceptors (abbreviated AFA) to define an abstract family of languages (abbreviated AFL) which is closed under reversal. This condition is satisfied by all of the well-known AFA which define reversal-closed AFL. A partial converse is given for AFL which are closed under both reversal and intersection with linear context-free languages.

# ON THE CLOSURE OF AFL UNDER REVERSAL*

## INTRODUCTION

In [5] the notion of an "abstract family of languages" (abbreviated AFL) was introduced as a model for many of the different families of languages of interest in automata and formal language theory. The notion of an "abstract family of acceptors" (abbreviated AFA) was then introduced as a model of the families of one-way nondeterministic acceptors. It was shown that a family of languages is accepted by an AFA if and only if it is an AFL closed under arbitrary homomorphism, and a family of languages is accepted by the "quasi-real-time" acceptors of an AFA if and only if it is an AFL containing the empty word. Thus the study of AFL and the study of AFA are closely related. It is therefore reasonable to impose properties on AFL and seek the corresponding properties on AFA. In the present note we study the property of reversal in an AFL. Specifically, we present a simple sufficiency condition on an AFA so that the associated AFL is closed under reversal. This condition is satisfied by all of the well-known AFA which define reversal-closed AFL. A partial converse is given for reversal-closed AFL which are closed under intersection with linear context-free languages.

*Research sponsored in part by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, USAF, Under contract F1962870C0023, the Air Force Office of Scientific Research, Office of Aerospace Research, USAF, under AFOSR Grant No. AF-AFOSR-1203-67A, and by the National Science Foundation, Grant No. GJ454.

## SECTION 1. PRELIMINARIES

In this section we recall some of the basic concepts involving families of languages and families of acceptors. We assume the reader is familiar with some of the elementary notions of sets of words such as concatenation, homomorphism, regular set, etc. All such undefined terms are in [5].

Definition. A family of languages is an ordered pair $(\Sigma,\mathcal{L})$, or $\mathcal{L}$ when $\Sigma$ is understood, where

(1) $\Sigma$ is an infinite set of symbols,

(2) $\mathcal{L}$ is a family of sets of words over $\Sigma$,

(3) For each L in $\mathcal{L}$ there is a finite set $\Sigma_1 \subseteq \Sigma$ such that $L \subseteq \Sigma_1^*$, and

(4) $L \neq \phi$ for some L in $\mathcal{L}$.

Henceforth, $\Sigma$ will always denote a given infinite set and $\Sigma$ subscripted a finite subset of $\Sigma$. Also, $\mathcal{L}$, with or without a subscript, will denote a family of languages (over $\Sigma$).

The special families of languages with which we shall be concerned are next defined.

Definition. An AFL (acronymn for "abstract family of languages") is a family $\mathcal{L}$ of languages closed under the operations of union, concatenation, +,[1] $\epsilon$-free homomorphism, inverse homomorphism, and intersection with regular sets. An AFL closed under every homomorphism is called a full AFL.

---

[1]Here, "+" is the Kleene closure operation without the empty word $\epsilon$. "*" is the Kleene closure operation with the empty word.

The notion of an AFL serves as a model for many of the important families of languages studied in automata and formal language theory. In particular, the regular sets, the context-free languages, the one-way nondeterministic stack languages, and the context-sensitive languages, each form an AFL, the first three also being full AFL.

The model for an abstract family of acceptors is rather complicated. It depends upon the following notion:

<u>Definition</u>. An AFA-schema is a 4-tuple $(\Gamma, I, f, g)$, with the following properties:

(1) $\Gamma$ and $I$ are abstract sets, with $\Gamma$ and $I$ nonempty.

(2) $f$ is a function from $\Gamma^* \times I$ into $\Gamma^* \cup \{\phi\}$.

(3) $g$ is a function from $\Gamma^*$ into the finite subsets of $\Gamma^*$ such that $g(\epsilon) = \{\epsilon\}$, and $\epsilon$ is in $g(\gamma)$ if and only if $\gamma = \epsilon$.

(4) For each $\xi$ in $g(\Gamma^*)$,[2] there is an element $1_\xi$ in $I$ satisfying $f(\gamma, 1_\xi) = \gamma$ for all $\gamma$ such that $g(\gamma)$ contains $\xi$.

(5) For each $u$ in $I$, there exists a finite set $\Gamma_u \subseteq \Gamma$ with the following property: If $\Gamma_1 \subseteq \Gamma$, $\gamma$ is in $\Gamma_1^*$, and $f(\gamma,u) \neq \phi$, then $f(\gamma,u)$ is in $(\Gamma_1 \cup \Gamma_u)^*$; that is, for each $\gamma$ in $\Gamma^*$, each symbol occurring in $f(\gamma,u)$ either occurs in $\gamma$ or is in $\Gamma_u$.

Intuitively, an AFA-schema is a type of auxiliary storage, with $g$ the "read" function and $f$ the "write" function. Elements of $\Gamma$ are auxiliary storage symbols, and elements of $I$ are "instructions." Further details and examples are in [5].

---

[2] For each set $A$, $g(A) = \bigcup_{\gamma \text{ and } A} g(\gamma)$.

Using the notion of an AFA-schema, we have the following concept:

<u>Definition</u>. An AFA (acronymn for "abstract family of one-way, nondeterministic acceptors") is a pair $(\Omega,\mathcal{D})$, or $\mathcal{D}$ when $\Omega$ is understood, with the following properties:

(1) $\Omega$ is a 6-tuple $(K,\Sigma,\Gamma,I,f,g)$, where

(a) $(\Gamma,I,f,g)$ is an AFA-schema, and

(b) $K$ and $\Sigma$ are infinite abstract sets.

(2) $\mathcal{D}$ is the family of all elements (called <u>acceptors</u>) $D = (K_1,\Sigma_1,\delta,q_0,F)$, where

(a) $K_1$ and $\Sigma_1$ are finite subsets of $K$ and $\Sigma$ resp., $F$ is a subset of $K_1$, and $q_0$ is in $K_1$, and

(b) $\delta$ is a function from $K_1 \times (\Sigma_1 \cup \{\epsilon\}) \times g(\Gamma^*)$ into the finite subsets of $K_1 \times I$ such that

$$G_D = \{\xi/\delta(q,a,\xi) \neq \phi \text{ for some } q \text{ and } a\}$$

is finite.

Thus an AFA is an AFA-schema together with all acceptors having the AFA-schema type of storage. Each acceptor has a finite number of states and input symbols. $q_0$ is the "start" state, and $F$ is the set of "accepting" states. $\delta$ is the "move" function. In order that an acceptor be finitely specified, $G_D$ is required to be finite.

An acceptor moves from configuration to configuration as follows:

<u>Notation.</u> Given an acceptor $D = (K_1,\Sigma_1,\delta,q_o,F)$, let $\vdash$ (or $\vdash_D$ when D is to be emphasized) be the relation on $K_1 \times \Sigma_1^* \times \Gamma^*$ defined as follows. For a in $\Sigma_1 \cup \{\epsilon\}$, $(p,aw,\gamma) \vdash (p',w,\gamma')$ if there exist $\xi$ and u such that $\xi$ is in $g(\gamma)$, $(p',u)$ is in $\delta(p,a,\xi)$, and $f(\gamma,u) = \gamma'$. Let $\overset{*}{\vdash}$ be the reflexive, transitive extension of $\vdash$.

We now define "acceptance" in an acceptor. Intuitively, a word w is accepted if the acceptor, starting from the start state with empty storage, reads all of w and ends in an accepting state with empty storage.

<u>Definition.</u> Let $(\Omega,\mathfrak{D})$ be an AFA and let $D = (K_1,\Sigma_1,\delta,q_o,F)$ be in $\mathfrak{D}$. Let L(D), called the <u>set</u> (or <u>language</u>) <u>accepted by D</u>, be the set of words

$$\{w \text{ in } \Sigma_1^* \mid (p_o,w,\epsilon) \overset{*}{\vdash} (p,\epsilon,\epsilon) \text{ for some } p \text{ in } F\}.$$

Let $\mathcal{L}(\mathfrak{D}) = \{L(D) \mid D \text{ in } \mathfrak{D}\}$.

It is shown in [5] that for each family $\mathcal{L}$ of languages, $\mathcal{L}$ is a full AFL if and only if there exists an AFA $\mathfrak{D}$ such that $\mathcal{L} = \mathcal{L}(\mathfrak{D})$.

We need one other concept about acceptors.

<u>Definition.</u> Let $\mathfrak{D}$ be an AFA and $k \geq 0$. Let $\mathfrak{D}_k^t$ be the set of all D in $\mathfrak{D}$ such that $(p_1,\epsilon,\gamma_1) \vdash \cdots \vdash (p_m,\epsilon,\gamma_m)$ implies $m \leq k$. Each D in $\bigcup_{k\geq 0} \mathfrak{D}_k^t$ is said to be a <u>quasi-real-time acceptor</u> and each L in $\mathcal{L}^t(\mathfrak{D}) = \bigcup_{k\geq 0} \mathcal{L}(\mathfrak{D}_k^t)$ a <u>quasi-real-time language.</u>

It is shown in [5] that for each family of languages $\mathcal{L}$, $\mathcal{L}$ is an AFL containing $\{\epsilon\}$ if and only if there exists an AFA $\mathfrak{D}$ such that $\mathcal{L} = \mathcal{L}^t(\mathfrak{D})$.

## SECTION 2. RESULTS

We are interested in the following operation:

Definition. Let $\epsilon^R = \epsilon$ and $(a_1 \ldots a_k)^R = a_k \ldots a_1$, each $a_i$ in $\Sigma_1$, $k \geq 1$. If $X \subseteq \Sigma_1^*$, let $X^R = \{x^R/x \text{ in } X\}$. The operation which maps x into $x^R$ and X into $X^R$ is called reversal.

We shall be concerned with the study of AFL $\mathcal{L}$ closed under reversal, i.e., if L is in $\mathcal{L}$, then $L^R$ is in $\mathcal{L}$. It was noted in [5] that the smallest AFL $\mathcal{L}$ containing the language $L_o = \{a^n b^m/0 \leq m < n\}$ is not closed under reversal since $\mathcal{L}$ does not contain $L_o^R$. A more "natural" example is the family $\mathcal{L}_N$ of one-way, nondeterministic, nonerasing stack-acceptor languages.[3] It was proved in [12] that $\mathcal{L}_N$ contains $L_1 = \{a^{n^2} b^n/n \geq 1\}$, but $\mathcal{L}_N$ does not contain $L_1^R$.

While an AFL $\mathcal{L}$ need not be closed under reversal, it does contain a unique, maximal AFL closed under reversal, namely $\mathcal{L} \cap \mathcal{L}^R = \{L/L \text{ and } L^R \text{ in } \mathcal{L}\}$, where $\mathcal{L}^R = \{L^R/L \text{ in } \mathcal{L}\}$.

The remainder of this paper concerns a condition on an AFA $\mathcal{D}$ which implies that $\mathcal{L}^t(\mathcal{D})$ and $\mathcal{L}(\mathcal{D})$ are closed under reversal. This condition is defined as follows:

Definition. Let $(\Omega, \mathcal{D})$ be an AFA, with $\Omega = (K, \Sigma, \Gamma, I, f, g)$. The AFA is said to be reversible if there exists a one to one function h from $\Gamma^*$ into $\Gamma^*$ satisfying the following conditions:

(1) $gh(\gamma) = hg(\gamma)$ for all $\gamma$ in $\Gamma^*$.

(2) $h(\epsilon) = \epsilon$.

---

[3] See [6] for the definition of these languages.

(3) For each u in I and $\xi$ in $g(\Gamma^*)$, there exists $v_{u,\xi}$ in I so that for each $\gamma$ and $\gamma'$, $\xi$ in $g(\gamma)$, $f(\gamma,u) = \gamma'$ if and only if $f(h(\gamma'),v_{u,\xi}) = h(\gamma)$.

This condition is related to, but different from, a condition stated in [9].

In all practical applications we know, h may be taken to be the identity function on $\Gamma^*$.

__Example__. Let K, $\Sigma$, and $\Gamma$ be infinite sets. Let $I = \{\epsilon, E_Z/Z \text{ in } \Gamma\} \cup \Gamma$, where $E_Z$ is a new symbol for each Z in $\Gamma$. Let $(\Omega,\mathfrak{A})$ be the AFA where $f(\gamma,\epsilon) = \gamma$, $f(\gamma Z,E_Z) = \gamma$, $f(\gamma,Z) = \gamma Z$, $g(\epsilon) = \{\epsilon\}$, and $g(\gamma Z) = \{Z\}$ for each $\gamma$ in $\Gamma^*$ and Z in $\Gamma$. Then $\mathfrak{A}$ is reversible, with h the identity function, and is the AFA of pushdown acceptors.

In a similar manner, it is easy to check that all of the following families are reversible AFA: nondeterministic finite-state acceptors with $\epsilon$-moves; nondeterministic one-counters [3]; nondeterministic one-way stack acceptors [6]; list-storage acceptors [7]; and nested stack acceptors [1]. On the other hand, in view of Theorem 2.1 below and because the family of one-way, nondeterministic, nonerasing, stack-acceptor languages are not closed under reversal, the AFA of one-way nondeterministic, nonerasing stack acceptors are not reversible. Speaking informally, there is no way to reverse the addition of a symbol to the stack.

The following result is our sufficiency condition for an AFL to be closed under reversal.

<u>Theorem 2.1.</u> $\mathcal{L}(\mathfrak{A})$ and $\mathcal{L}^t(\mathfrak{A})$ are both closed under reversal for each reversible AFA $\mathfrak{A}$.

<u>Proof.</u> Let L be in $\mathcal{L}(\mathfrak{A})$. Then there exists an acceptor $D_1 = (K_1,\Sigma_1,\delta_1,p_1,F_1)$ in $\mathfrak{A}$ such that $L = L(D)$. Without loss of generality, we may assume that $F_1 = \{d\}$ and $\delta_1(d,a,\xi) = \phi$ for each a and $\xi$. For each $(u,\xi)$ in $I \times g(\Gamma^*)$, let [4]

$$H_{u,\xi} = [gf(g^{-1}(\xi),u)] \cap G_{D_1} = \{\xi'/\delta_1(q,a,\xi') \neq \phi \text{ for some } q,\ a, \text{ and}$$
$$\xi',\xi' \text{ in } g(f(\gamma,u)),\ \xi \text{ in } g(\gamma) \text{ for some } \gamma\}.$$

Since $G_{D_1}$ is finite, $H_{u,\xi}$ is finite (possibly empty). Let $D_2$ be the acceptor $(K_2,\Sigma_1,\delta_2,(d,\epsilon),\ \{(p_1,\epsilon)\})$, where $K_2 = K_1 \times G_{D_1}$ and $\delta_2$ is defined as follows: If $(p,u)$ is in $\delta_1(q,a,\xi)$, then let $((q,h(\xi)),v_{u,\xi})$ be in $\delta_2((p,h(\xi')),\ a,\ h(\xi'))$ for all $\xi'$ in $H_{u,\xi}$.

Since $K_1$ and $G_{D_1}$ are both finite, $K_2$ is finite and $D_2$ is an acceptor. The fact that $L(D_2) = (L(D_1))^R$ and the fact that $D_1$ is in $\mathfrak{A}_k^t$ for some k if and only if $D_2$ is in $\mathfrak{A}_k^t$ for some k is an immediate consequence of the following.

Let $k \geq 1$ and $w_1,\ldots,w_k$ be in $\Sigma_1 \cup \{\epsilon\}$. Let $q_o,\ldots,q_k$ be in K and $\gamma_o,\ldots,\gamma_k$ in $\Gamma^*$. Then

(1) $(q_o,w_k\cdots w_1,\gamma_o) \vdash_{D_1} (q_1,w_{k-1}\cdots w_1,\gamma_1) \vdash_{D_1} \cdots \vdash_{D_1} (q_k,\epsilon,\gamma_k)$, with $g(\gamma_k) \neq \phi$,

if and only if

(2) $((q_k,h(\xi_k)),w_1\cdots w_k,h(\gamma_k)) \vdash_{D_2} \cdots \vdash_{D_2} ((q_o,h(\xi_o)),\epsilon,h(\gamma_o))$ for some $\xi_o$ in $g(\gamma_o)$, $\ldots$, $\xi_k$ in $g(\gamma_k)$.

---

[4] $g^{-1}(\xi) = \{\gamma \text{ in } \Gamma^*/\xi \text{ is in } g(\gamma)\}$.

To see that (1) implies (2), we use induction on k. The case for $k = 1$ is trivial and is subsumed in the case $k = m + 1$ given below. Suppose (1) implies (2) for $k \le m$. Consider $k = m + 1$. Suppose (1) holds for $k = m + 1$. Then we have

(3) $(q_0, w_{m+1}, \gamma_0) \vdash_{D_1} (q_1, \epsilon, \gamma_1)$ and

(4) $(q_1, w_m \cdots w_1, \gamma_1) \vdash_{D_1} \cdots \vdash_{D_1} (q_{m+1}, \epsilon, \gamma_{m+1})$,

with $g(\gamma_1) \neq \phi$ and $g(\gamma_{m+1}) \neq \phi$. From (4) and induction,

(5) $((q_{m+1}, h(\xi_{m+1})), w_1 \cdots w_m, h(\gamma_{m+1})) \overset{*}{\vdash}_{D_2} ((q_1, h(\xi_1)), \epsilon, h(\gamma_1))$,

with $\xi_i$ in $g(\gamma_i)$, $1 \le i \le m + 1$. From (3), there exist $\xi_0$ in $g(\gamma_0)$ and $(q_1, u_1)$ in $\delta_1(q_0, w_{m+1}, \xi_0)$ such that $f(\gamma_0, u_1) = \gamma_1$. Then $\xi_1$ is in $H_{u_1, \xi_0}$. By construction of $D_2$ we have

(6) $((q_0, h(\xi_0)), v_{u_1, \xi_0})$ is in $\delta_2((q_1, h(\xi_1)), w_{m+1}, h(\xi_1))$.

By definition of reversibility,

(7) $f(h(\gamma_1), v_{u_1, \xi_0}) = h(\gamma_0)$.

Since $\xi_1$ is in $g(\gamma_1)$,

(8) $h(\xi_1)$ is in $hg(\gamma_1) = gh(\gamma_1)$.

From (6), (7), and (8), we have

(9) $((q_1, h(\xi_1)), w_{m+1}, h(\gamma_1)) \vdash_{D_2} ((q_0, h(\xi_0)), \epsilon, h(\gamma_0))$.

Combining (5) and (9), we get (2) for $k = m + 1$.

Using induction we now show that (2) implies (1). Suppose $k = 1$. Then

(10) $((q_1,h(\xi_1)),\ w_1,\ h(\gamma_1)) \vdash_{\overline{D_2}} ((q_o,h(\xi_o)),\ \epsilon,\ h(\gamma_o))$

for some $\xi_o$ in $g(\gamma_o)$ and some $\xi_1$ in $g(\gamma_1)$. Then

(11) $((q_o,h(\xi_o)),v)$ is in $\delta_2((q_1,h(\xi_1)),w_1,\bar{\xi})$ and

(12) $f(h(\gamma_1),v) = h(\gamma_o)$

for some $\bar{\xi}$ in $gh(\gamma_1)$ and some v in I. By construction of $\delta_2$, $\bar{\xi} = h(\xi_1)$. Since h is one to one, $\xi_o,\gamma_o,\xi_1$, and $\gamma_1$ are uniquely defined from $h(\xi_o)$, $h(\gamma_o)$, $h(\xi_1)$, and $h(\gamma_1)$. By (11) and the definition of $\delta_2$, $v = v_{u,\xi_o}$ for some u in I satisfying (3) in the definition of reversibility such that

(13) $(q_1,u)$ is in $\delta_1(q_o,w_1,\xi_o)$.

From (12) and from (3) in the definition of reversibility,

(14) $f(\gamma_o,u) = \gamma_1$.

From (13), (14), $\xi_o$ in $g(\gamma_o)$, and $\xi_1$ in $g(\gamma_1)$, we have

(15) $(q_o,w_1,\gamma_o) \vdash_{\overline{D_1}} (q_1,\epsilon,\gamma_1)$, with $g(\gamma_1) \neq \phi$,

i.e., (1) holds.

Assume (2) implies (1) for $1 \leq k \leq m$ and suppose $k = m + 1$. Then

(16) $((q_{m+1},h(\xi_{m+1})),\ w_1,\ h(\gamma_{m+1})) \vdash_{\overline{D_2}} ((q_m,h(\xi_m)),\epsilon,h(\gamma_m))$ and

(17) $((q_m,h(\xi_m)),\ w_2\cdots w_{m+1},\ h(\gamma_m)) \vdash^{*}_{\overline{D_2}} ((q_o,h(\xi_o)),\ \epsilon,\ h(\gamma_o))$

for some $\xi_o$ in $g(\gamma_o),\ldots,\xi_{m+1}$ in $g(\gamma_{m+1})$. By induction, we get

(18) $(q_m,w_1,\gamma_m) \vdash_{\overline{D_1}} (q_{m+1},\epsilon,\gamma_{m+1})$ and

(19) $(q_o,w_{m+1}\cdots w_2,\gamma_o) \vdash^{*}_{\overline{D_1}} (q_m,\epsilon,\gamma_m)$,

with $g(\gamma_{m+1}) \neq \phi$. Hence $(q_o,w_{m+1}\cdots w_1,\gamma_o) \vdash^{*}_{\overline{D_1}} (q_{m+1},\epsilon,\gamma_{m+1})$, with $g(\gamma_{m+1}) \neq \phi$, i.e., (1) holds, thereby completing the proof.

From the theorem, we immediately obtain another proof of the known fact that each of the following AFL is closed under reversal: the regular sets; the nondeterministic one-counter languages; the nondeterministic one-way (quasi-real-time) stack-acceptor languages; the list-storage acceptor languages; and the (quasi-real-time) nested stack-acceptor languages.

While Theorem 2.1 is not difficult to prove, it should be useful in eliminating machine proofs of closure under reversal for new families of languages defined by AFA.

We had hoped to be able to characterize AFL closed under reversal by AFA. Unfortunately, we have not been successful in that we can only give a partial converse to Theorem 2.1.

Theorem 2.2. Let $\mathcal{L}$ be a (full) AFL which contains $\{\epsilon\}$, is closed under reversal, and is closed under intersection with linear context-free languages.[5] Then there exists an AFA $\mathfrak{D}$ which is reversible such that $\mathcal{L} = \mathcal{L}^t(\mathfrak{D})$ $(\mathcal{L} = \mathcal{L}(\mathfrak{D}))$.

Proof. Let K be an infinite set. For each element a in $\Sigma$ let $a'$, $a''$, and $a'''$ be new symbols. For each language L in $\mathcal{L}$ let $e_L$ and $\bar{e}_L$ be new symbols. Let 1 be a new symbol. For each a and b in $\Sigma$, let $E_{(a',b')}$ and $E_{(a''',b''')}$ be new symbols. Let $\Sigma' = \{a'/a \text{ in } \Sigma\}$, $\Sigma'' = \{a''/a \text{ in } \Sigma\}$, and $\Sigma''' = \{a'''/a \text{ in } \Sigma\}$.

---

[5] A linear context-free language is a language generated by a context-free grammar in which all products are of the form $\xi \rightarrow w$ or $\xi \rightarrow w_1 \nu w_2$, where $\xi$ and $\nu$ are variables and w, $w_1$ and $w_2$ are words over the terminal-letter alphabet. See [4].

Since $\Sigma$ is infinite, we may assume that $\Sigma \times \Sigma$, $\Sigma' \times \Sigma'$, $\Sigma'' \times \Sigma''$, and $\Sigma''' \times \Sigma'''$ are pairwise disjoint subsets of $\Sigma$. Let

$\Gamma = (\Sigma \times \Sigma) \cup (\Sigma' \times \Sigma') \cup (\Sigma'' \times \Sigma'') \cup (\Sigma''' \times \Sigma''')$, and

$I = (\Sigma \times \Sigma) \cup (\Sigma'' \times \Sigma'') \cup \{e_L, \bar{e}_L / L \text{ in } \mathcal{L}\} \cup \{1, E_{(a',b')}, E_{(a''',b''')} / a,b \text{ in } \Sigma\}$.

Let $g$ be the function on $\Gamma^*$ defined by $g(\epsilon) = \{\epsilon\}$ and $g(\gamma Z) = \{Z\}$ for all $\gamma$ in $\Gamma^*$ and $Z$ in $\Gamma$. Let $f$ be the function from $\Gamma^* \times I$ into $\Gamma^* \cup \{\phi\}$ defined as follows (for each $\gamma$ in $\Gamma^*$, $x$ in $(\Sigma \times \Sigma)^*$, $x'$ in $(\Sigma' \times \Sigma')^*$, $x''$ in $(\Sigma'' \times \Sigma'')^*$, $x'''$ in $(\Sigma''' \times \Sigma''')^*$, $k \geq 1$, $M$ and $N$ in $\mathcal{L}$, and $a$, $a_i$, $b$, $b_i$ in $\Sigma$):

(1) $f(\gamma,1) = \gamma$.

(2) $f(x,(a,b)) = x(a,b)$.

(3) $f((a_1,b_1)\dots(a_k,b_k),e_M) = (a_1',b_1')\dots(a_k',b_k')$ if $a_1\dots a_k b_k \dots b_1$ is in $M$.

(4) $f(x'(a',b'),E_{(a',b')}) = x'$.

(5) $f(x'',(a'',b'')) = x''(a'',b'')$.

(6) $f((a_1'',b_1'')\dots(a_k'',b_k''),\bar{e}_N) = (a_1''',b_1''')\dots(a_k''',b_k''')$ if $b_1\dots b_k a_k \dots a_1$ is in $N$.

(7) $f(x'''(a''',b'''),E_{(a''',b''')}) = x'''$.

(8) $f = \phi$ in all other instances.

Let $(\Omega,\mathfrak{L})$ be the AFA for which $\Omega = (K,\Sigma,\Gamma,I,f,g)$. Let $h$ be the isomorphism on $\Gamma^*$ generated by $h((a,b)) = (a''',b''')$, $h((a'',b'')) = (a',b')$, $h((a',b')) = (a'',b'')$, and $h((a''',b''')) = (a,b)$ for each $a$ and $b$ in $\Sigma$. It is easily verified that $\mathfrak{L}$ is reversible since (1) and (1), (2) and (7), (3) and (6), and (4) and (5) are "reverses" of each other. Note that (3) and (6) are reverses of one another if and only if $N = M^R$. Thus the fact that $\mathcal{L}$ is closed under reversal is implicitly used.

To complete the proof it suffices to show that $\mathcal{L} = \mathcal{L}^t(\mathcal{D})$. (An analogous proof shows $\mathcal{L} = \mathcal{L}(\mathcal{D})$.) Consider $\mathcal{L} \subseteq \mathcal{L}^t(\mathcal{D})$. To prove this containment, it suffices to show that

(9) Each L in $\mathcal{L}$ consisting only of even-length words is in $\mathcal{L}^t(\mathcal{D})$.

[For let L be any language in $\mathcal{L}$ and $L \subseteq \Sigma_L^*$. Then $L = L_e \cup L_o$, where $L_e = L \cap (\Sigma_L^2)^*$ and $L_o = L \cap \Sigma_L(\Sigma_L^2)^*$. Since $\mathcal{L}$ is an AFL, $L_e$, the even-length words in L, is in $\mathcal{L}$ and $L_o$, the odd-length words, is in $\mathcal{L}$. For each a in $\Sigma_L$, let $L_{oa} = L_o \cap a\Sigma_L^*$. Then $L_o = \bigcup_a L_{oa}$ and each $L_{oa}$ is in $\mathcal{L}$. Since $\mathcal{L}$ is an AFL containing $\{\epsilon\}$, each set $a\backslash L_{oa} = \{w/aw \text{ in } L_{oa}\}$ is in $\mathcal{L}$ [5]. Furthermore, each $a\backslash L_{oa}$ contains only even-length words. By (9), $L_e$ and each $a\backslash L_{oa}$ is in $\mathcal{L}^t(\mathcal{D})$. Thus $L = L_e \cup \bigcup_a a(a\backslash L_{oa})$ is in the AFL $\mathcal{L}^t(\mathcal{D})$.]

Hence let L in $\mathcal{L}$ be a set containing only even-length words. Let $p_o$ and $p_1$ be two symbols in K. Let $F_L = \{p_o,p_1\}$ if $\epsilon$ is in L and $F_L = \{p_1\}$ if $\epsilon$ is not in L. Let $D_L$ be the acceptor $(\{p_o,p_1\},\Sigma_L,\delta_L,p_o,F_L)$, where $\delta_L$ is defined by $\delta_L(p_o,a,\xi) = \{(p_o,(a,b))/b \text{ in } \Sigma_L\}$ for each a in $\Sigma_L$ and $\xi$ in $(\Sigma_L \times \Sigma_L) \cup \{\epsilon\}$, $\delta_L(p_o,\epsilon,\xi) = \{(p_1,e_L)\}$ for each $\xi$ in $\Sigma_L \times \Sigma_L$, and $\delta_L(p_1,b,(a',b')) = \{(p_1,E_{(a',b')})\}$ for each (a,b) in $\Sigma_L \times \Sigma_L$. Obviously $L(D_L) = L$, so that L is in $\mathcal{L}^t(\mathcal{D})$.

We now show that $\mathcal{L}^t(\mathcal{D}) \subseteq \mathcal{L}$. The proof of this containment is more complicated and uses the hypothesis about the closure of $\mathcal{L}$ under intersection with linear context-free languages. Let $D = (K_1,\Sigma_o,\delta,p_o,F)$ be an acceptor in $\mathcal{D}$. We may assume that if (q,u) is in $\delta(p,x,\xi)$ and $\xi = (a',b')$, then u is either 1 or $E_{(a',b')}$. [Otherwise, the acceptor blocks.] Write $(p,w,\epsilon) \overset{-}{\vdash} (q,y,\gamma)$ if either

(10) $(p,w,\epsilon) = (q,y,\gamma)$, or

(11) $(p,w,\epsilon) \vdash (q,y,\gamma)$, or

(12) There exist $n \geq 1$, $p_1,\ldots,p_n$, $\gamma_1,\ldots,\gamma_n$, $w_1,\ldots,w_n$ such that

(a) $(p,w,\epsilon) \vdash (p_1,w_1,\gamma_1)$,

(b) $(p_i,w_i,\gamma_i) \vdash (p_{i+1},w_{i+1},\gamma_{i+1})$ for each $i$ such that $1 \leq i \leq n$,

(c) $(p_n,w_n,\gamma_n) \vdash (q,y,\gamma)$, and

(d) $\gamma_i \neq \epsilon$ for all $i$, $1 \leq i \leq n$.

For each $p$ and $q$ in $K_1$, let $\alpha(p,q)$ be a distinct symbol in $\Sigma - \Sigma_o$ and

$$L_{pq} = \{w \text{ in } \Sigma_O^*/(p,w,\epsilon) \overset{*}{\vdash} (q,\epsilon,\epsilon)\}.$$

Let $\Sigma_1 = \{\alpha(p,q)/p,q \text{ in } K_1\}$ and let $R$ be the regular set

$$R = \{\alpha(p_o,p)/p \text{ in } F\} \cup \{\alpha(p_o,p_{i_1})\alpha(p_{i_1},p_{i_2})\ldots\alpha(p_{i_n},p)/n \geq 1, p_i \text{ in } K_1,\ p \text{ in } F\}.$$

Let $\tau_1$ be the substitution [6] on $\Sigma_1^*$ defined by $\tau_1(\alpha(p,q)) = L_{pq}$ for each $\alpha(p,q)$. Clearly $L(D) = \tau_1(R)$. Since each AFL containing $\{\epsilon\}$ is closed under substitution into regular sets by languages in $\mathcal{L}$ [5], it suffices to show that each $L_{pq}$ is in $\mathcal{L}$.

---

[6] Let $\Sigma_1$ be a finite set and $\mathcal{L}$ a family of languages. For each $a$ in $\Sigma_1$ let $\tau(a)$ be a language in $\mathcal{L}$. Let $\tau$ be the function on $\Sigma_1^*$ defined by $\tau(\epsilon) = \{\epsilon\}$ and $\tau(x_1\ldots x_k) = \tau(x_1)\ldots\tau(x_k)$ for each $k \geq 1$, $x_i$ in $\Sigma_1$. Then $\tau$ is called a <u>substitution</u> (on $\Sigma_1^*$). For each $X \subseteq \Sigma_1^*$ let $\tau(X) = \bigcup_{x \text{ in } X} \tau(x)$. Then $\tau(X)$ is called a <u>substitution into X by languages of $\mathcal{L}$.</u>

Let $I_D = \{u/(p,u)$ in $\delta(q,a,\xi)$ for some $q, a, \xi\}$ and $\mathcal{E}_D = I_D \cap \{e_M, \bar{e}_M/M$ in $\mathcal{L}\}$. For each p and q in $K_1$ and each e in $\mathcal{E}_D$, let $L^e_{pq}$ be the set of all words w in $\Sigma_o^*$ with the following property: There exist $n \geq 2$ and appropriate $(p_i,w_i,\gamma_i)$, $1 \leq i \leq n$, such that $(p_1,w_1,\gamma_1) \vdash \cdots \vdash (p_n,w_n,\gamma_n)$, $w = w_1$, $w_n = \epsilon$, $p_1 = p$, $p_n = q$, $\gamma_1 = \gamma_n = \epsilon$, $\gamma_k \neq \epsilon$ for all k, $1 < k < n$, and $(p_{j+1},e)$ is in $\delta(p_j,w_j,g(\gamma_j))$, with $f(\gamma_j,e) = \gamma_{j+1}$, for some j. Intuitively, $L^e_{pq}$ is the set of input words which cause D to leave state p with empty storage and go into state q with empty storage, without emptying the storage sometime in the "interior" of the computation. Moreover, the instruction e, and no other instruction in $\mathcal{E}_D$, is used, and exactly once, during the sequence of moves. [In fact, disregarding the instances when the instruction 1 occurs, the instruction e occurs at the "midpoint" of the computation.] Now a word w is in $L_{pq}$ if and only if either (i) $w = \epsilon$ and $p = q$, or (ii) w is in $\Sigma_o \cup \{\epsilon\}$ and $(p,w,\epsilon) \vdash (q,\epsilon,\epsilon)$ by an application of the instruction 1, or (iii) w causes D to leave state p with empty storage and, in at least two moves, go into state q with empty storage, with all intermediate storages empty. Since (iii) can occur only if an instruction in $\mathcal{E}_D$ is used in the computation, we have

$$(13) \quad L_{pq} = \bigcup_{e_M \text{ in } \mathcal{E}_D} L^{e_M}_{pq} \cup \bigcup_{\bar{e}_M \text{ in } \mathcal{E}_D} L^{\bar{e}_M}_{pq} \cup T,$$

where T is a finite subset of $\Sigma_o \cup \{\epsilon\}$. Since $\mathcal{E}_D$ is finite, (13) is a finite union. Since an AFL is closed under union and T is in $\mathcal{L}$, it suffices to show that each $L^{e_M}_{pq}$ and each $L^{\bar{e}_M}_{pq}$ is in $\mathcal{L}$.

We now show that each $L^{e_M}_{pq}$ is in $\mathcal{L}$, an analogous argument holding for $L^{\bar{e}_M}_{pq}$. Let M, p, and q be given. Let

$$L_1 = \{(a_1,b_1)\dots(a_k,b_k)(a_k',b_k')\dots(a_1',b_1')/k \geq 1, \text{ each } (a_i,b_i) \text{ in } \Sigma_o \times \Sigma_o\}.$$

Then $L_1$ is a linear context-free language over $\Sigma$ since it is generated by the grammar whose set of productions is

$$\{\sigma \to (a,b)\ \sigma(a',b'),\ \sigma \to (a,b)(a',b')/(a,b) \text{ in } \Sigma_o \times \Sigma_o\}.$$

Let $\tau_2$ be the substitution on $\Sigma_o^*$ defined by $\tau_2(a) = \{(a,b),(b',a')/b \text{ in } \Sigma_o\}$. Each set $\tau_2(a)$ is finite and does not contain $\epsilon$. Therefore $\tau_2(M)$ is in $\mathcal{L}$ [5]. Let $L_M = L_1 \cap \tau_2(M)$. Since $\mathcal{L}$ is closed under intersection with linear context-free languages, $L_1 \cap \tau_2(M)$ is in $\mathcal{L}$. It is easily seen that

$$L_M = \{(a_1,b_1)\dots(a_k,b_k)(a_k',b_k')\dots(a_1',b_1')/k \geq 1, \text{ each } (a_i,b_i)$$
$$\text{in } \Sigma_o \times \Sigma_o, a_1\dots a_k b_k \dots b_1 \text{ in } M\}.$$

For each instruction $u \neq 1$ in $I_D$, let $1_u$ be a new symbol. For each $s$, $t$, $a$, $\xi$, and $u$ such that $(t,u)$ is in $\delta(s,a,\xi)$, let $(s,a,\xi,t,u)$ be a new symbol in $\Sigma$ if $u \neq 1$ and $(s,a,\xi,t,1_v)$ be a new symbol in $\Sigma$ for each $1_v$ if $u = 1$. Let $\Sigma_2$ be the set of all the 5-tuples. Clearly $\Sigma_2$ is finite. Let $W$ be the set of all words in $\Sigma_2^*$ of the form

$$(p_1,a_1,\xi_1,p_2,u_1)(p_2,a_2,\xi_2,p_3,u_2)\dots(p_m,a_m,\xi_m,p_{m+1},u_m),$$

where $m \geq 2$, $p_1 = p$, $p_{m+1} = q$, $\xi_1 = \epsilon$, $\xi_i \neq \epsilon$ for $1 < i \leq m$; $u_m$ is of the form $E_{(a',b')}$, and for each $j$, $1 \leq j < m$; (i) $u_j = (a,b)$ implies $u_{j+1} = (c,d)$, $e_M$, or $1_{u_j}$, (ii) $u_j = e_M$ implies $u_{j+1} = E_{(a',b')}$ or $u_{j+1} = 1_{e_M}$, (iii) $u_j = E_{(a',b')}$ implies $u_{j+1} = E_{(c',d')}$ or $u_{j+1} = 1_{E_{(a',b')}}$, and (iv) $u_j = 1_v$ implies $u_{j+1}$ is given by (i), (ii), or (iii), according as v is of the form (a,b), $e_M$, or $E_{(a',b')}$. Then W is the set of all computations, in coded form, in which D leaves state p with empty storage and goes to state q, without emptying the storage in the interior of the computation, using the instruction $e_M$ exactly once, and using no other instruction in $\mathcal{E}_D$. Note that the computation in W need not end with empty storage. The role of $1_u$ is to indicate the use of Instruction 1 while remembering the last non-1 instruction. Clearly W is a regular set.

To complete the proof that $L_{pq}^{e_M}$ is in $\mathcal{L}$, we shall construct homomorphisms $h_1$ and $h_2$, with $h_1$ $\epsilon$-limited on [7] $h_2^{-1}(L_M) \cap W$, such that

(14) $L_{pq}^{e_M} = h_1[h_2^{-1}(L_M) \cap W]$.

It is known [5] that if $\mathcal{L}$ is an AFL containing $\{\epsilon\}$ and $h_1$ is an $\epsilon$-limited homomorphism on U, U in $\mathcal{L}$, then $h_1(U)$ is in $\mathcal{L}$. Since an AFL is closed under inverse homomorphism and intersection with regular sets, it will follow that $L_{pq}^{e_M}$ is in $\mathcal{L}$. Hence let $h_1$ be the homomorphism on $\Sigma_2^*$ generated by $h_1((s,a,\gamma,t,u)) = a$ for each element $(s,a,\gamma,t,u)$ in $\Sigma_2$. Let $h_2$ be the homomorphism on $\Sigma_2^*$ generated by $h_2((s,a,\gamma,t,(c,d))) = (c,d)$ for each $(s,a,\gamma,t,(c,d))$ in $\Sigma_2$, $h_2((s,a,(c',d')),$ $t,E_{(c',d')})) = (c',d')$ for each $(s,a,(c',d'),t,E_{(c',d')})$ in $\Sigma_2$ , and $h_2(y) = \epsilon$ for all other elements y in $\Sigma_2$. It is a straightforward matter to verify that (14) holds. Note that each word in $h_2^{-1}(L_M) \cap W$ is a

---

[7] A homomorphism h is <u>$\epsilon$-limited on a language U</u> if there exists $k \geq 0$ such that for all w in U, if w = xyz and $h(y) = \epsilon$, then the length of y is less than k.

computation in coded form which empties the storage. If D is in $\mathcal{D}_k^t$ for some $k \geq 0$, then obviously $h_1$ is $\epsilon$-limited on W and thus on $h_2^{-1}(L_M) \cap W$. Hence the theorem.

Consider the hypothesis to Theorem 2.2 for $\mathcal{L}$ an AFL. Most families of languages defined by "natural" families of two-way acceptors contain the linear context-free languages and are closed under intersection and reversal. (Exceptions exist, such as the family of nondeterministic finite-state acceptors with $\epsilon$-input moves.) It was shown in [8] that the family $\mathcal{L}$ of languages defined by a family of two-way acceptors becomes an AFL, $\mathcal{F}(\mathcal{L})$, when closed under $\epsilon$-free homomorphism. If $\mathcal{L}$ is closed under reversal and intersection with linear context-free languages, then the same is true of $\mathcal{F}(\mathcal{L})$. [For suppose h is an $\epsilon$-free homomorphism, L is in $\mathcal{L}$, and $L_o$ is linear context-free. Then $h(L)^R = h'(L^R)$, where $h'$ is the homomorphism generated by $h'(a) = (h(a))^R$ for each a. Since $h^{-1}(L_o)$ is also linear, $h(L) \cap L_o = h(L \cap h^{-1}(L_o))$ is in $\mathcal{F}(\mathcal{L})$.] Thus $\mathcal{F}(\mathcal{L})$ satisfies the hypotheses of Theorem 2.2 for families $\mathcal{L}$ of languages defined by most families of two-way acceptors. Also, families of languages defined by most "natural" families of one-way nondeterministic quasi-real-time multistorage tape acceptors satisfy the hypotheses of Theorem 2.2 [10].

Consider the hyoptheses to Theorem 2.2 for $\mathcal{L}$ a full AFL. Note that $\mathcal{L}$ is closed under intersection.[8] [For let $L_1$ and $L_2$ be in $\mathcal{L}$, with $L_1 \subseteq \Sigma_1^*$ and $L_2 \subseteq \Sigma_1^*$. Let c be a new symbol in $\Sigma$. Since $L_3 = L_1 c \Sigma_1^* c L_2$ is in $\mathcal{L}$

---

[8] We are indebted to Dr. Ronald Book for this observation.

and $L_4 = \{wcw^Rcy/w$ and $y$ in $\Sigma_1^*\}$ is linear context-free, $L_5 = L_3 \cap L_4 = \{wcw^Rcy/w$ in $L_1, y$ in $L_2\}$ is in $\mathcal{L}$. Since $L_6 = \{ycw^Rcw/w$ and $y$ in $\Sigma_1^*\}$ is linear context-free, $L_5 \cap L_6 = \{wcw^Rcw/w$ in $L_1 \cap L_2\}$ is in $\mathcal{L}$. From the fact that $\mathcal{L}$ is a full AFL, it then follows that $L_1 \cap L_2$ is in $\mathcal{L}$.] Also, $\mathcal{L}$ contains $\{a^nb^n/n \geq 1\}$. Now the smallest full AFL containing $\{a^nb^n/n \geq 1\}$ and closed under intersection is the recursively enumerable sets [11]. Thus any full AFL satisfying the hypotheses of Theorem 2.2 contains the recursively enumerable sets.

Finally, let $\Sigma_1 \subseteq \Sigma$ and for each a in $\Sigma_1$ let $a'$ be a new symbol in $\Sigma$, with $\Sigma_1' = \{a'/a$ in $\Sigma_1\}$. Let $L(\Sigma_1,\Sigma_1')$ be the linear context-free language $\{a_1 \ldots a_k a_k' \ldots a_1'/k \geq 1, a_i$ in $\Sigma_1\}$. An examination of the proof of Theorem 2.2 reveals that it is only necessary to assume that

(*) $L \cap L(\Sigma_1,\Sigma_1')$ is in $\mathcal{L}$ for each $L$ in $\mathcal{L}$ and each $L(\Sigma_1,\Sigma_1')$.

For the full AFL case, however, it is easily seen that (*) implies that $\mathcal{L}$ is closed under intersection with arbitrary linear context-free languages. For let $L_1$ be an arbitrary linear context-free language. It is noted in [2] that there exists a regular set $U$ and homomorphisms $h_1$ and $h_2$ such that $L_1 = \{h_1(w)\, h_2(w^R)/w$ in $U\}$. Let $U \subseteq \Sigma_1^*$. Then

$$L_1 = h_3(L(\Sigma_1,\Sigma_1') \cap U\Sigma_1'^*),$$

where $h_3$ is the homomorphism on $(\Sigma_1 \cup \Sigma_1')^*$ generated by $h_3(a) = h_1(a)$ and $h_3(a') = h_2(a)$ for each a in $\Sigma_1$. Hence

$$
\begin{aligned}
L \cap L_1 &= L \cap h_3(L(\Sigma_1,\Sigma_1') \cap U\Sigma_1'^{*}) \\
&= h_3[h_3^{-1}(L) \cap L(\Sigma_1,\Sigma_1') \cap U\Sigma_1'^{*}] \\
&= h_3[(h_3^{-1}(L) \cap U_1\Sigma_1'^{*}) \cap L(\Sigma_1,\Sigma_1')],
\end{aligned}
$$

which is in $\mathfrak{L}$.

BIBLIOGRAPHY

[1] Aho, A.V., "Nested Stack Automata," Journal of the Association for Computing Machinery, Vol. 16 (1969), pp. 383-406.

[2] Chomsky, N. and M.P. Schutzenberger, "The Algebraic Theory of Context-Free Languages," in P. Braffort and D. Hirschberg (eds.), Computer Programming and Formal Systems, Amsterdam: North Holland Publishing Co., 1963, pp. 118-161.

[3] Fischer, P.C., "Turing Machines with Restricted Memory Access," Information and Control, Vol. 9 (1966), pp. 364-379.

[4] Ginsburg S., The Mathematical Theory of Context-Free Languages, New York: McGraw-Hill, 1966.

[5] Ginsburg, S. and S. Greibach, "Abstract Families of Languages," in Studies in Abstract Families of Languages, Memoir of the American Mathematical Society, Number 87 (1969), pp. 1-32.

[6] Ginsburg, S., S. Greibach, and M.A. Harrison, "One-Way Stack Automata," Journal of the Association for Computing Machinery, Vol. 14 (1967), pp. 389-418.

[7] Ginsburg, S., and M.A. Harrison, "One-Way Nondeterministic Real-Time List-Storage Languages," Journal of the Association for Computing Machinery, Vol. 15 (1968), pp. 428-446.

[8] Ginsburg, S. and J. Hopcroft, "Two-Way Balloon Automata and AFL," to appear in the Journal of the Association for Computing Machinery.

[9] Greibach, S., Full AFL's and Nested Iterated Substitutions, Harvard University Report, 1969.

[10] Greibach, S. and S. Ginsburg, Multitape AFA , SDC document TM-738/050/00, January 2, 1969.

[11] Hartmanis, J. and J. Hopcroft, What Makes Some Language Theory Problems Decidable, Cornell University Report, 1969.

[12] Ogden, W., "Intercalation Theorems for Stack Languages," ACM Symposium on Theory of Computing, Marina del Rey, California, May 5-7, 1969.

UNCLASSIFIED

Security Classification

**DOCUMENT CONTROL DATA - R & D**

*(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)*

| | |
|---|---|
| 1. ORIGINATING ACTIVITY (Corporate author)<br>System Development Corporation<br>2500 Colorado Blvd.<br>Santa Monica, California 90406 | 2a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED<br>2b. GROUP |

3. REPORT TITLE

ON THE CLOSURE OF AFL UNDER REVERSAL

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

"SCIENTIFIC. INTERIM"

5. AUTHOR(S) (First name, middle initial, last name)

Seymour Ginsburg

Michael Harrison

| | | |
|---|---|---|
| 6. REPORT DATE<br>10 November 1969 | 7a. TOTAL NO. OF PAGES<br>26 | 7b. NO. OF REFS<br>12 |

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO. F1962870C0023,<br>b. Project, Task, Work Unit Nos.<br>5632-05-01<br>c. DoD Element: 61102F<br>d. DoD Subelement: 681305 | 9a. ORIGINATOR'S REPORT NUMBER(S)<br>TM-738/056/00<br>SCIENTIFIC REPORT NO. 2<br>9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)<br>AFCRL-69-0551 |

10. DISTRIBUTION STATEMENT

1-This document has been approved for public release and sale; its distribution is unlimited.

| | |
|---|---|
| 11. SUPPLEMENTARY NOTES<br>Air Force Office of Scientific Research<br>Office of Aerospace Research, USAF<br>Arlington, Virginia | 12. SPONSORING MILITARY ACTIVITY<br>Air Force Cambridge Research Laboratories (CRB)<br>L.G. Hanscom Field<br>Bedford, Massachusetts 01730 |

13. ABSTRACT

A simple sufficiency condition is given for an abstract family of acceptors (abbreviated AFA) to define an abstract family of languages (abbreviated AFL) which is closed under reversal. This condition is satisfied by all of the well-known AFA which define reversal-closed AFL. A partial converse is given for AFL which are closed under both reversal and intersection with linear context-free languages.

| 14 KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Abstract family of acceptors (AFA) | | | | | | |
| Abstract family of languages (AFL) | | | | | | |
| Acceptors | | | | | | |